

# Browser Widget

The browser widget allows you to display and interact with a web page within your app. It wraps browser views on all platforms using native browser objects on Mac, iOS and Android, and using the Chromium Embedded Framework (CEF) on Windows and Linux.



## Creating a Browser

A browser widget can be created by dragging it out from the Tools Palette, where it appears with the following icon:



Alternatively it can be created in script using:

```
create widget as "com.livecode.widget.browser"
```

## Displaying Content in a Browser

The most straightforward way to use the browser widget is to set its `url` property to display a web page. It can also be set to a local file url.

Alternatively the `htmlText` property can be used to render specific chunks of html within the browser object.

## Controlling Navigation

The browser sends the following messages which can be used to control navigation within the browser. See the API documentation for more details:

- browserDocumentLoadBegin
- browserDocumentLoadComplete
- browserDocumentLoadFailed
- browserFrameDocumentLoadBegin
- browserFrameDocumentLoadComplete
- browserFrameDocumentLoadFailed
- browserNavigateBegin
- browserNavigateComplete
- browserNavigateFailed
- browserUnhandledLoadRequest

## Interacting With JavaScript

The browser allows two-way interaction with the javascript running in the displayed page. To allow LiveCode handlers to be called from JavaScript, there is a `javascriptHandlers` property. This can be set to a list of LiveCode handlers which appear as methods attached to a global `liveCode` object. You can call these methods as you would any other JavaScript function and pass whatever parameters you require.

In the other direction, you can use the `do in widget` command to execute JavaScript statements in a browser widget. For example:

```
-- Use JavaScript to hide "myButton" in the page displayed in a browser widget.  
do "document.getElementById('myButton').hidden = 'hidden'" in widget "myBrowser"
```